First Name

Last Name

First let
of last na

UW Userid

# E&CE 327 Midterm

2012t2 (Spring)

## Instructions and General Information

- 100 marks total. Time limit: 1 hour, 20 minutes (80 minutes)
- No books, no notes, no computers. Calculators are allowed
- If you need extra paper, request some from a proctor.
- Write neatly. To earn partial credit, you must show the formulas you are using and all of your work.
- **The proctors and instructors will not answer questions, except in cases where an error on the exam is suspected. If you are confused about a question, write down your assumptions or interpretation.**
- **Justifications of answers will be marked according to correctness, clarity, and concision.**

| | | | Total Marks | Approx. Time | Page |
|---|---|---|---|---|---|
| Q0 | !!Almost Free!! | | 2 | 2 | 2 |
| Q1 | Short Answer | | 25 | 20 | 3 |
| Q2 | VHDL Simulation | | 20 | 15 | 5 |
| Q3 | The Good, the Bad, and the Unsynthesizable | | 20 | 15 | 7 |
| Q4 | State Machines | | 20 | 15 | 11 |
| Q5 | Dataflow Diagram with Memory | | 15 | 10 | 13 |
| Totals | | | 100 | 77 | |

# Q0 (2 Marks) !!Almost Free!!

(*estimated time: 2 minutes*)

## Q0a (1 Mark) Best part

What is the best part of the course?

## Q0b (1 Mark) Most improve

What one thing could be done to most improve the course for the remainder of the term?

# Q1 (25 Marks) Short Answer

(*estimated time: 20 minutes*)

## Q1a (4 Marks) VHDL Semantics

What are the two defining characteristics of zero-delay simulation, and how does VHDL simulation achieve these characteristics?

Characteristic 1: _____

How does VHDL achieve this characterisic? _____

Characteristic 2: _____

How does VHDL achieve this characterisic? _____

## Q1b (7 Marks) Idle State

When is an idle state needed in a control table?

What could go wrong if you do not include an idle state when one is needed?

## Q1c (7 Marks) Unused

When must the "unused" symbol be used in a control table?

What could go wrong if you do not use the "unused" symbol when one is needed?

## Q1d (7 Marks) State Encoding

Which state encoding is probably the best choice for a system that has a *latency of 3 clock cycles* and where there are *at least 2 bubbles* between each pair of valid parcels?

Either write down the name of the encoding, or describe the encoding. If you write the name of the encoding, you do not need to describe it.

# Q2 (20 Marks) VHDL Simulation

(*estimated time: 15 minutes*)

For the VHDL program below, how many *simulation cycles* will occur in the simulation round that begins at 10ns?

Answer the question by using the list on the next to page to give a **brief** description of what happens in each simulation cycle of the simulation round.

```
entity simple is
end entity;

architecture main of simple is
  signal clk, a, b, c, d, e, f, g : std_logic;
begin
  proc1 : process begin
    clk <= '0';
    wait for 10 ns;
    clk <= '1';
    wait for 10 ns;
  end process;
  proc2 : process begin
    a <= '0';
    b <= '1';
    wait for 15 ns;
    a <= '1';
    wait for 20 ns;
  end process;
  proc3 : process begin
    wait until rising_edge(clk);
    c <= a;
  end process;
  proc4 : process (a, c, e) begin
    d <= a xor c;
    f <= not e;
  end process;
  proc5 : process (d) begin
    e <= d;
  end process;
  proc6 : process begin
    wait until rising_edge(clk);
    g <= f;
  end process;
end architecture;
```

**NOTES:**

1. Your descriptions should be brief: do *not* include all of the information that is in a waveform diagram. The goal of this question is to evaluate your understanding of VHDL simulation semantics without all of the tedious details of the full waveform diagram.

| Simulation cycle count | Description |
|---|---|
| 1: | |
| 2: | |
| 3: | |
| 4: | |
| 5: | |
| 6: | |
| 7: | |
| 8: | |

**Scratch work area**

# Q3 (20 Marks) The Good, the Bad, and the Unsynthesizable

(*estimated time: 15 minutes*)

**NOTES:**

1. For each of the code fragments Q3a–Q3e, answer whether the code is legal VHDL.
2. If the code is legal VHDL, answer whether it is synthesizable.
3. If the code is synthesizable:
   (a) Answer whether it adheres to good coding practices, according to the guidelines for E&CE 327.
   (b) Draw the hardware that would be most likely to result from synthesizing the code.
   (c) If the VHDL code includes an implicit state machine: draw the gates, wires, and flops for the datapath. You may draw the control portion of the circuit as a cloud or black-box that drives the appropriate signals in the datapath. Draw all of the connections between the implicit state machine and the datapath, inputs, and outputs.
4. If the the code is not legal, or is not synthesizable, or does not follow good coding practices, explain why.
5. All signals are `std_logic` and are internal to an architecture.
6. You do not need to draw the signal `clk` if it is used as the clock input to a flip-flop. You *must* draw the signal `clk` if it is used for *something other than a clock input to a flip-flop in the datapath*.
7. If a signal other than `clk` is used as the clock-input to flip-flop in the datapath, then you must draw that signal.

**Q3a**

```
process (clk) begin
  if rising_edge(clk) then
    d0 <= z;
    d1 <= a;
    if b = '1' then
      z <= d1;
    else
      z <= d0;
    end if;
  end if;
end process;
```

| | Yes | No |
|---|---|---|
| Legal | ☐ | ☐ |
| Synthesizable | ☐ | ☐ |
| Good Practice | ☐ | ☐ |

Explanation: _____

Drawing of hardware, if synthesizable:

**Q3b**

```
process (clk) begin
  if rising_edge(clk) then
    d0 <= z;
    d1 <= a;
  end if;
end process;
process begin
  wait until rising_edge(clk);
  if b = '1' then
    z <= d1;
  else
    z <= d0;
  end if;
end process;
```

| | Yes | No |
|---|---|---|
| Legal | ☐ | ☐ |
| Synthesizable | ☐ | ☐ |
| Good Practice | ☐ | ☐ |

Explanation: _____

Drawing of hardware, if synthesizable:

**Q3c**

```
process (clk) begin
  if rising_edge(clk) then
    if b = '1' then
      z <= a;
    end if;
  end if;
end process;
```

| | Yes | No |
|---|---|---|
| Legal | ☐ | ☐ |
| Synthesizable | ☐ | ☐ |
| Good Practice | ☐ | ☐ |

Explanation: _____

Drawing of hardware, if synthesizable:

**Q3d**

```
process (clk) begin
  if rising_edge(clk) then
    d0 <= z;
    d1 <= a;
    s  <= b;
  end if;
end process;

if s = '1' generate
  z <= d1;
end generate;
if s = '0' generate
  z <= d0;
end generate;
```

| | Yes | No |
|---|---|---|
| Legal | ☐ | ☐ |
| Synthesizable | ☐ | ☐ |
| Good Practice | ☐ | ☐ |

Explanation: _____

Drawing of hardware, if synthesizable:

**Q3e**

```
process begin
  wait until rising_edge(clk);
  d0 <= z;
  d1 <= a;
  wait until rising_edge(clk);
  if b = '1' then
    z <= d1;
  else
    z <= d0;
  end if;
end process;
```

| | Yes | No |
|---|---|---|
| Legal | ☐ | ☐ |
| Synthesizable | ☐ | ☐ |
| Good Practice | ☐ | ☐ |

Explanation: _____

Drawing of hardware, if synthesizable:

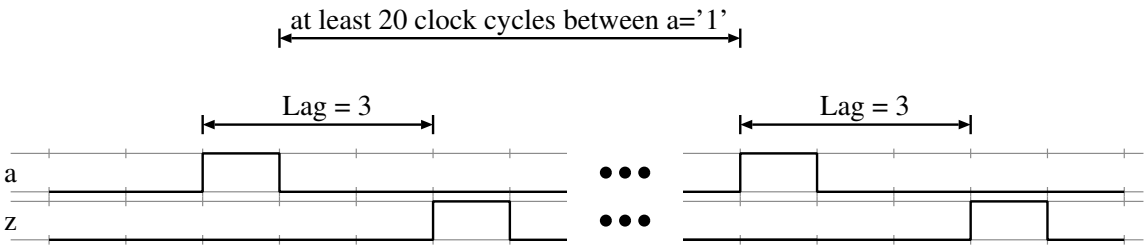
# Q4 (20 Marks) State Machines

(*estimated time: 15 minutes*)

The intended behaviour of the state machines below is that whenever the input `a='1'`, the machine will generate `z='1'` some number of clock cycles later (See waveform below). The number of clock cycles between `a='1'` and `z='1'` is called the "lag" of the system. Each state machine may have a different lag.

**NOTES:**

1. For each of the state machines Q4a–Q4c, answer whether the state machine is correct with respect to the intended behaviour.
2. If the state machine is correct, answer what its lag is.
3. If the state machine is incorrect, explain either how the machine's behaviour differs from the intended behaviour or how the state machine could be modified to fix the incorrect behaviour.
4. The state machines are allowed to ignore the value of `a` in the first clock cycle.
5. The input `a` is guaranteed to be `'0'` for at least 20 clock cycles between each pair of clock cycles when `a` is `'1'`.

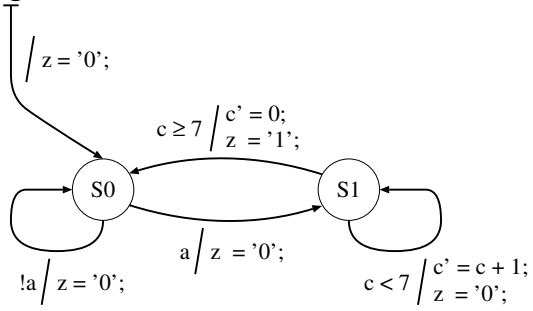



**Example waveforms with a lag of 3 clock cycles**

---

**The problem continues on the next page**

**Q4a**



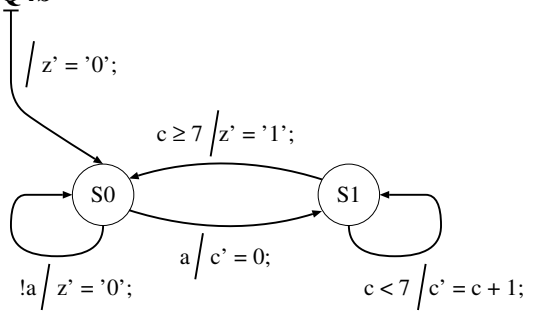

Yes No

Machine is correct ☐ ☐

If correct: lag ☐

If incorrect, explanation:

**Q4b**



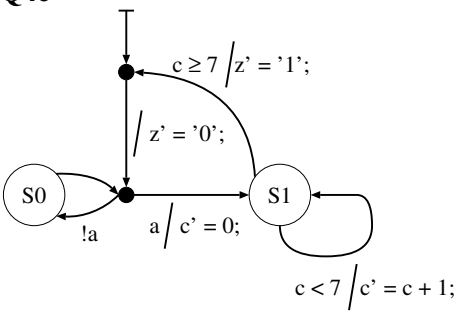

Yes No

Machine is correct ☐ ☐

If correct: lag ☐

If incorrect, explanation:

**Q4c**




Yes No

Machine is correct ☐ ☐

If correct: lag ☐

If incorrect, explanation:

## Q5 (15 Marks) Dataflow Diagram with Memory

(*estimated time: 10 minutes*)

Draw a dataflow diagram that implements the specification:

```
M[a-1] = d;
z      = M[a] + M[a-1];
```

**NOTES:**

1. Inputs shall be *registered*
2. Outputs shall be *combinational*
3. Memory has registered inputs and combinational outputs (same as in class)
4. The memory is *single-ported*
5. There are exactly *five bubbles* between each pair of consecutive valid parcels.
6. Optimization goals in order of decreasing importance:
   (a) minimize *area*
      i. registers (excluding memory)
      ii. total of (number of adders + number of subtracters)
      iii. input ports
      iv. output ports
   (b) minimize *latency*
7. Input values may be read in any clock cycle, but each input value shall be read exactly once.
8. Optimizations to the pseudocode are allowed, so long as the values on `z` and `M` are correct.